

---

**TECHNICKÁ UNIVERZITA V LIBERCI**  
**Fakulta mechatroniky, informatiky a mezioborových studií**

**Studijní program:** B2612 Elektrotechnika a informatika

**Studijní obor:** 2612R011 Elektronické informační a řídicí systémy



**Tvorba ukázkových aplikací průmyslových  
robotů v prostředí ABB Robot Studio**

**Creating Demo Applications of Industrial  
Robots in ABB Robot Studio Software**

**Bakalářská práce**

Autor: Jan Hnyk  
Vedoucí: Ing. David Lindr, Ph.D.  
Konzultant: Ing. Josef Černohorský, Ph.D.

vakat



**TECHNICKÁ UNIVERZITA V LIBERCI**  
**Fakulta mechatroniky, informatiky a mezioborových studií**  
Akademický rok: 2012/2013

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan Hnyk**  
Osobní číslo: **M09000057**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronické informační a řídicí systémy**  
Název tématu: **Tvorba ukázkových aplikací průmyslových robotů v prostředí ABB Robot Studio**  
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Seznamte se s technickými prostředky automatizační techniky firmy ABB a prostředím ABB Robot Studio používaným pro programování průmyslových robotů.
2. Teoretické poznatky získané studiem technické dokumentace a odborné literatury aplikujte při tvorbě ukázkových aplikací prezentujících možnosti spolupráce průmyslových robotů firmy ABB.
3. Vypracujte závěrečnou textovou dokumentaci bakalářské práce, zhodnoťte výsledky vaší práce a do elektronických příloh uveďte mimo jiné i zdrojové kódy vytvořených programů.



Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30–40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Záda, V.: Robotika, matematické aspekty analýzy a řízení, TUL, 2012, ISBN 978-80-7372-882-3.
- [2] Craig, J. J.: Introduction to Robotics Mechanics and Control, Third Edition, Prentice Hall, 2005.
- [3] RAPID Reference Manual, Article number: 3HAC 0966-13, Issue: For BaseWare OS 3.1 Rev.1, ABB Robotics Products AB, Sweden.

Vedoucí bakalářské práce: Ing. David Lindr, Ph.D.

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: Ing. Josef Černohorský, Ph.D.

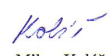
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: 10. října 2012

Termín odevzdání bakalářské práce: 17. května 2013

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2012



## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 — školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL, v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis



## Poděkování

Rád bych poděkoval vedoucímu mé Bakalářské práce Ing. Davidu Lindrovi, Ph.D. za trpělivost, odborný dohled a věcné připomínky.



## Abstrakt

Tato bakalářská práce si klade za cíl vytvořit ukázkové aplikace průmyslových robotů od firmy ABB. Úlohy jsou programovány v softwaru RobotStudio a následně jsou odsimulovány na průmyslových robotech. Konkrétně se jedná o tři aplikace, z níž první demonstruje opakovatelnost robota IRB140, který staví a následně rozebírá pyramidu z kostek. Druhá aplikace je zaměřená na komunikaci dvou robotů, kdy první je ve funkci podavače a druhý kostky odebírá a staví z nich útvar. Poslední úloha má demonstrovat přímé programování s nejnovějším systémem IRC5, kdy robot IRB120 manipuluje s křídou a kreslí po tabuli. Cílem bylo vytvořit návody a zajímavé úlohy, které budou sloužit pro výuku. Jednotlivé zdrojové kódy a návody budou proto volně k dispozici a budou sloužit jako podpora předmětu *Základy robotiky*, vyučované na fakultě mechatroniky TU v Liberci.

**Klíčová slova:** simulace, průmyslový robot, programování, RobotStudio, řídicí systém.



## Abstract

The main aim of this thesis is to create prototypical applications of the robots by ABB. RobotStudio software was used for the programming and then the simulation has been applied on the industrial robots. There are three applications tough. The first one demonstrates the repeatability of IRB140 robot, which is able to build up and then deconstruct the cube pyramid. The second application is focused on the communication between two robots, where the first one passes the cubes to the second one, which builds up the formation from them. The third application demonstrates the online programming with the latest IRC5 system, where the robot IRB120 is able to draw with chalk on a board. The objective was to create manual and interesting tasks useful for the further education. Individual source codes and tasks available and can be further used as a support for the classes of Fundamentals of Robotics, Faculty of Mechatronics at TU Liberec.

**Keywords:** simulation, industrial robot, programming, RobotStudio, control system.





# Obsah

Prohlášení	5
Poděkování	6
Abstrakt	7
Abstract	8
Obsah	10
Seznam obrázků	11
<b>1 Úvod</b>	<b>12</b>
<b>2 Robot</b>	<b>13</b>
2.1 Vývoj . . . . .	13
2.2 Průmyslová robotika . . . . .	14
2.3 Kinematika . . . . .	16
2.3.1 Angulární robot . . . . .	17
2.4 Řízení . . . . .	19
2.4.1 Přímá úloha . . . . .	19
2.4.2 Inverzní úloha . . . . .	19
2.5 Programování . . . . .	20
2.5.1 Dálkové (FlexPendant) . . . . .	20
2.5.2 Offline (RobotStudio) . . . . .	21
2.5.3 Online . . . . .	21
<b>3 Pracovní stanoviště</b>	<b>22</b>
3.1 Automatizační technika ABB . . . . .	22
3.1.1 Roboty řady IRB . . . . .	23
3.1.2 Řídicí systém . . . . .	24
3.2 Stavebnice ENGEL . . . . .	25



<b>4</b>	<b>Software</b>	<b>26</b>
4.1	ABB RobotStudio . . . . .	26
4.1.1	Prostředí . . . . .	27
4.1.2	Virtuální laboratoř . . . . .	27
4.1.3	RAPID . . . . .	29
<b>5</b>	<b>Úlohy</b>	<b>31</b>
5.1	Pyramida . . . . .	31
5.1.1	Kostky . . . . .	31
5.1.2	Úložná krabice a kalibrační podložka . . . . .	32
5.1.3	Příruba . . . . .	32
5.1.4	Prisavka1 . . . . .	33
5.1.5	Program . . . . .	34
5.2	Spolupráce . . . . .	36
5.2.1	Výrobky . . . . .	37
5.2.2	Efektory . . . . .	37
5.2.3	Program . . . . .	38
5.2.4	Problémy . . . . .	39
5.3	Tabule . . . . .	40
5.3.1	Stojan . . . . .	40
5.3.2	Chapadlo . . . . .	40
5.3.3	Offline . . . . .	41
5.3.4	Online . . . . .	41
	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>
	A Obsah CD . . . . .	45



## Seznam obrázků

1	Automatická kreslicí figurka . . . . .	13
2	Šest stupňů volnosti . . . . .	16
3	Angulární kinematická struktura . . . . .	18
4	Technické specifikace robota IRB120 [9] . . . . .	18
5	Dálkový ovladač . . . . .	20
6	Ukázka komunikace . . . . .	21
7	Rozložení robotů na pracovišti . . . . .	24
8	Stavebnice ENGEL . . . . .	25
9	Prostředí RobotStudio . . . . .	26
10	Model laboratoře . . . . .	27
11	Parametr ZONEDATA . . . . .	30
12	Barevné rozložení kostky . . . . .	31
13	Krabice . . . . .	32
14	Spojovací příruba . . . . .	33
15	Přísavka pro IRB140 . . . . .	34
16	Nastavení FTP . . . . .	35
17	Kalibrování pozic . . . . .	36
18	Detail propojení I/O . . . . .	37
19	Detail Přísavka2 . . . . .	38
20	Diagram předání kostky . . . . .	39
21	Chapadlo FESTO . . . . .	41

## Seznam tabulek

1	Souřadice modelů pracovního stanoviště . . . . .	28
---	--	----



# 1 Úvod

„...mladý Rossum měl nápad udělat z toho živé a inteligentní pracovní stroje...”

Karel Čapek

Svět okolo nás se rozvíjí nesmírnou rychlostí vpřed. Téměř každý den se setkávám s novou technologií či inovací na poli techniky. Pro člověka v oboru je velkou výzvou držet krok s okolním světem a neupadnout v neznalost. Od dob, kdy člověk sestrojil první stroje určené k hromadnému usnadnění práce uběhla již dlouhá doba. Dnes je běžná robotizace celého výrobního procesu. Člověk již není v pozici dělníka, ale stává se z něj osoba dohlížející na chod stroje, který provádí těžkou nebo stereotypní práci.

V této práci se zabývám analýzou robotického pracoviště a tvorbou ukázkových aplikací v softwaru ABB Robot Studio. Výsledky budou sloužit jako studijní materiál a programy, které zde budou uvedeny jako názorná ukázka možností práce s průmyslovými roboty pro předmět Základy robotiky, vyučované na Technické univerzitě v Liberci pod fakultou mechatroniky, informatiky a mezioborových studií. V jednotlivých kapitolách se budu věnovat stručnému seznámení s automatizační technikou a se softwarem, ve kterém budu pracovat. Uvedené postupy a návody jsou psány stručně, ale zároveň jsem se snažil návody tvořit tak, aby byly srozumitelné i pro začátečníky. Zdokumentované aplikace budou volně přístupné, aby si je případný student mohl bezproblému vyzkoušet.

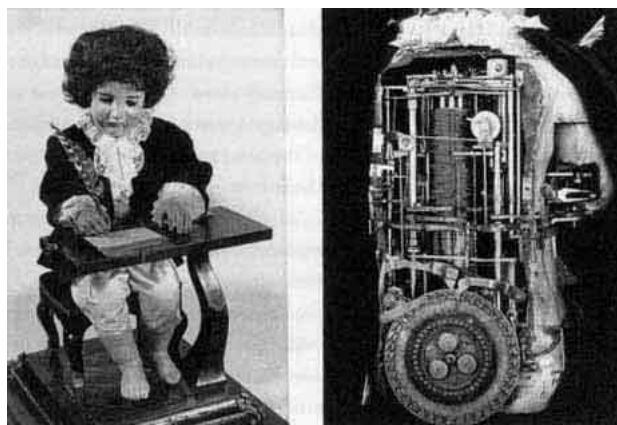


## 2 Robot

Slovo „robot“ bylo poprvé v dějinách použito v roce 1920 ve hře R.U.R - *Rossum's Universal Robots* Karla Čapka. Karel Čapek přesto není vynálezcem tohoto slova. Je jím jeho bratr Josef Čapek, který mu tento výraz poradil když se s bratrem bavili o hře, kterou měl Karel Čapek v plánu napsat [2].

### 2.1 Vývoj

Od nepaměti se člověk zabýval otázkou mechanizace a automatizace svého okolí. Zpočátku byly aplikace založené na čistě mechanickém principu. Za zmínku stojí některé vynálezy Leonarda da Vinci nebo automaty, které sestrojili Pier a Henry Droz. V 18. století se postupně vytvářelo mnoho těchto mechanických „hraček“. <sup>1</sup>



Obrázek 1: Automatická kreslicí figurka

K vývoji robotů velmi přispěl rozvoj elektrotechniky. Na počátku 20. století je zaznamenán veliký skok. Objevují se praktické aplikace, které značně usnadňují stereotypní nebo fyzicky náročnou práci. Jsou to různé teleoperátory, které manipulují s nebezpečnými materiály. Objevují se numericky řízené obráběcí stroje a výrobní linky. V roce 1961 je po vývoji na Columbia University uveden do provozu první průmyslový robot UNIMATE, který financovala firma General Motors. Ta robota využila pro výrobu automobilů, konkrétně se jednalo o vyjmutí a manipulaci s horkým a těžkým odlitkem. Objevuje se zde nový vědní obor, který

<sup>1</sup>Obrázek, <https://eee.uci.edu/clients/bjbecker/RevoltngIdeas/jacquetdrozwriterb.jpg>, read:(13.1.2013)



nazýváme *umělá inteligence* (UI). Základem jsou obecné zákonitosti v pozadí základních kognitivních procesů (tj. procesů vnímání a racionálního myšlení). K těmto procesům přispívá neustálý rozvoj sensoriky, kdy můžeme robotům přiřadit vidění za pomoci nejrůznějších čidel a kamer. Takto vybavený robot může analyzovat velmi dobře svoje prostředí a tím i reagovat na případné změny v jeho okolí. V literárních zdrojích je slovo robot definováno různě. Abychom pokryli univerzálnost robotů, se kterými se můžeme v dnešní době setkat, definujeme robota jako počítačem řízený integrovaný systém, schopný autonomní a cílově orientované interakce s reálným prostředím v souladu s instrukcemi od člověka. Každý robot musí též splňovat základní zákony robotiky tak, jak je definoval spisovatel Issac Asimov již v r. 1950 v knize Já robot (I, Robot)[3].

- Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.
- Robot musí uposlechnout příkazy člověka, kromě případů, kdy tyto příkazy jsou v rozporu s prvním zákonem.
- Robot musí chránit sám sebe před zničením, kromě případů, kdy tato ochrana je v rozporu s prvním nebo druhým zákonem.

## 2.2 Průmyslová robotika

Ve 20. století se zejména v automobilovém průmyslu objevují výrobní linky jako hlavní prostředek zintenzivnění velkovýroby. Začlo se využívat různých prostředků automatizace výroby včetně jednoúčelových mechanismů. Například uchopovací a podávací zařízení. Tento vývoj vedl k nasazení jednoduchých manipulátorů s přiřazeným řídicím systémem. V dnešní době jsou roboty součástí výrobního procesu mnoha firem po celém světě. Jejich finanční a časové přínosy pro výrobu jsou nezanedbatelné. Dalšími důvody jsou malá poruchovost, vysoká přesnost a zejména rychlá opakovatelnost činnosti. Nejčastěji využívanými typy robotů v průmyslu jsou:

- Šestiosé manipulátory (angulární struktura). Je to nejčastěji aplikovaná varianta. Díky počtu os nachází uplatnění v naprosté většině průmyslových aplikací.
- Paletizační roboty (čtyř-osá struktura). Jsou navrženy pro pohyby bez nutnosti reorientace břemene. Hlavními přednostmi jsou vysoká rychlost a dosah.



- Paralerní struktury (tricept). Předností tohoto konceptu je velmi vysoká rychlost v úzkém akčním rádiu. Nejčastější využití je ve farmaceutickém, potravinářském průmyslu a při montáži solárních článků.
- Portálové varianty (gantry). Výhodou těchto konceptů je široký pracovní rozsah. S vlastním pohonem se dokáží přizpůsobit velkému pracovišti. Najdeme je u obsluhy obráběcích center a při paletizaci.
- Lakovací roboty. Nejblíže mají k angulárním robotům. Jediný rozdíl je v tom, že je zde potřeba maximální hybnost na koncové části robota, kde je umístěna lakovací pistole.
- Scara struktura. Dříve často využívaná v elektronickém průmyslu pro svoji rychlost. Dnes je nahrazována angulární variantou, která neustále zvyšuje své rychlosti.

Tyto struktury nacházejí široké uplatnění v různých průmyslových odvětvích. Roboty jsou využívány jako jeden z významných nástrojů výroby. Nejčastěji se s robotizací pracoviště setkáváme v:

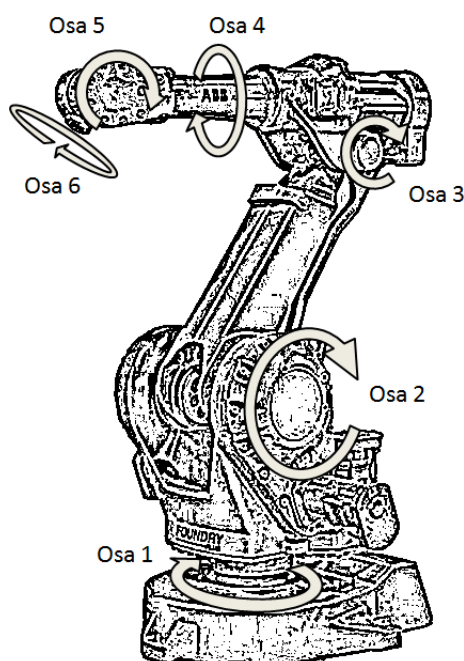
- Potravinářském průmyslu, kde je robotika uplatňována při paletizaci, balení finálních výrobků nebo při třídění zboží pomocí struktury tricept.
- Strojírenském průmyslu, kde je potřeba obsluhovat kovárenské stroje. Za zmínku stojí aplikace přesného lití nebo obsluha kovacího procesu. Roboty pro obrábění, lisování, svařování, montáž a balení.
- Elektrotechnickém průmyslu, kde je potřeba manipulovat s těžkými výrobky a polotovary, vyjímání výlisků, osazování DPS a celková kompletace výrobků.
- Plynárenský, Hornický a Stavebnický průmysl. Roboty zde zastupují obrovské těžkopádné stroje, jako jsou rypadla, jeřáby, portálové manipulátory.
- Armádě, která využívá roboty pro práci s výbušninami nebo pro práci v zamořeném prostředí. A pro vojenské nasazení ve formě bojových robotů a bezpilotních letadel.
- Zdravotnických aplikací. Tyto aplikace umožňují složité operace srdce, mozku, očí nebo šití tenkých cév.



V automobilovém průmyslu se setkáme s propojením mnoha aplikací využití robotizace. Využívá se obloukové svařování karoserií, lakování jednotlivých dílů, lisování plastů, následná montáž a především finální testování produktů. Stručně mohu konstatovat, že robotika zasáhla všechna průmyslová odvětví. Díky ní jsou výrobní procesy rychlejší, přesnější a mnohdy kvalitnější. Nehledě na to, že robot šetří místo na pracovišti a přebírá neoblíbené, nebezpečné či stereotypní práce. Roboty „milují“ práci, kterou lidé nenávidí.

## 2.3 Kinematika

Mechanickou konstrukci průmyslového robota tvoří pohybový systém, který je rozdělen na dvě části. O nastavení polohy v prostoru se stará *polohovací systém* a o natočení a následnou orientaci objektu v prostoru se stará *orientační systém*. Koncový člen robota najdeme na výstupu orientačního ústrojí a běžně na něj umísťujeme efektor. Efektor slouží jako nástavba koncového členu, který umožňuje manipulaci s objekty, obloukově svařovat, lakovat, apod. O tom jaký má robot pracovní prostor rozhoduje uspořádání a typy vazeb mezi jednotlivými kinematickými dvojicemi. V praxi se nejčastěji rozlišují dva základní typy kinematických dvojic.



Obrázek 2: Šest stupňů volnosti





- Translační T (Posuv): pohyb se realizuje po vytyčené úsečce. Jedná se o lineární posuv tělesa vůči druhému. Využívá se suportový, teleskopický nebo smykadlový mechanismus. Jak je uvedeno v [13, kapitola 2.1.2. na straně 7.].
- Rotační R (Otočení): umožňuje dvěma tělesům, které jsou k sobě takto vázány otočný pohyb kolem jedné osy kartézského souřadného systému. Poskytuje dvěma tělesům ve vzájemné vazbě jeden ze šesti možných stupňů volnosti.

Celkový počet stupňů volnosti mechanismu je dán jeho konstrukcí a počtem stupňů volnosti jednotlivých kinematických dvojic. Na obrázku č.2 je znázorněna konstrukce robota se šesti stupni volnosti.

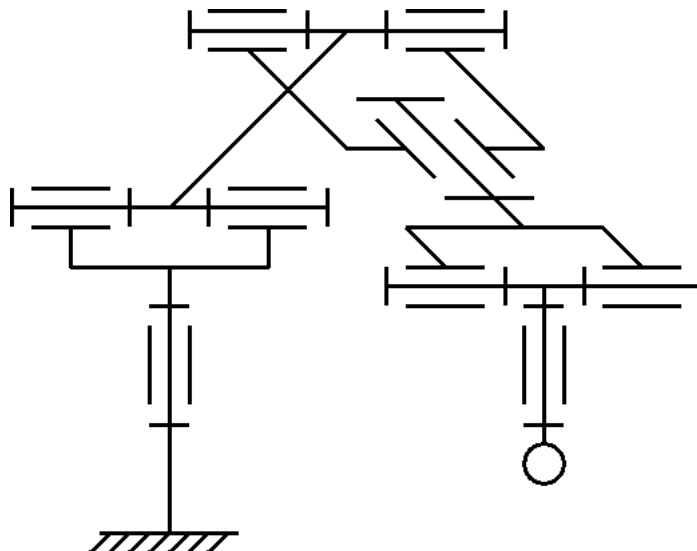
S ohledem na konstrukci rozdělujeme čtyři základní typy kinematických struktur. Každá z uvedených struktur má své jedinečné vlastnosti a tím je vhodná pro určitý typ úloh. Celek pak tvoří tzv. *základní kinematický řetězec*. Rozdělení do jednotlivých kategorií je provedeno na základě typů prvních třech os, které jsou počítány od základny robota. Robot se šesti stupni volnosti většinou poslední tři osy využívá k určení orientace tělesa v prostoru. Mezi základní typy kinematických struktur patří:

- kartézská (pravoúhlá) struktura - TTT: robot dokáže vykonávat lineární posuvy v jednotlivých osách. Největší výhody tohoto uspořádání je stabilita a nosnost. Využívají se pro jednoduché montáže nebo jsou používány jako skladové systémy.
- cylindrická (válcová) struktura - TRT: pracovní prostor robota je tvořen válcem. Zpravidla umožňuje rychlou a přesnou manipulaci. Nalezneme je napříč celým spektrem průmyslových odvětví, nejčastěji však jako různé paletizátory.
- sférická (kulová) struktura - RRT: pracovní prostor tvoří segment duté koule. Uplatňuje se v aplikacích, kde není zapotřebí velká nosnost a daleký rozsah ramene.
- angulární (antropomorfní) - RRR: nejrozšířenější koncepce, která je blíže popsána v následující podkapitole.

### 2.3.1 Angulární robot

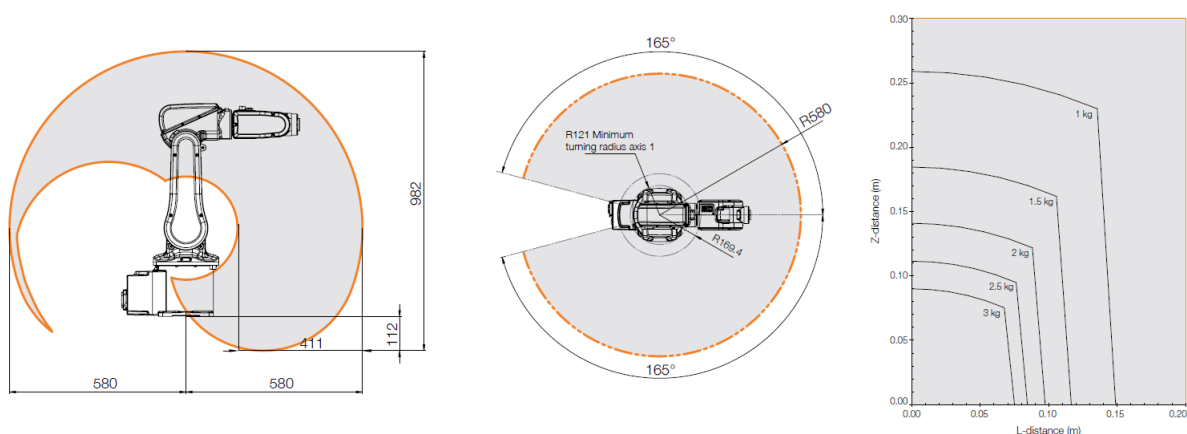
Angulární robot využívá pouze rotačních pohybů. Ovšem je možné rozšířit podstavec o pojezdovou plochu a tím rozšířit pracovní prostor o jeden translační pohyb. V praxi má

tato struktura často šest stupňů volnosti, které zařizují všestranost při aplikacích. Na obrázku č.3 je znázorněná kinematická struktura angulárního robota. Díky struktuře RRR má robot velký pracovní prostor.



Obrázek 3: Angulární kinematická struktura

Na obrázku č.4 je znázorněno zatížení, dosah a pracovní prostor robota IRB120 od společnosti ABB. Ikdyž je nejmenší ve své řadě, jeho pracovní poloměr je postačujících 580 mm. Tyto aspekty získané z produktové dokumentace [9] určují pro jaké aplikace bude robot určen. Jednotlivé parametry jsou rozebrány v kapitole 3.1 na straně 22.



Obrázek 4: Technické specifikace robota IRB120 [9]



## 2.4 Řízení

Při řízení průmyslových robotů se využívají dvě kinematické úlohy. Buď se řeší polohy kloubů nebo koncového členu robota. Jak uvedl ve své knize Václav Záda [[5, str. 80, kap. (4.5. Přímá a inverzní úloha robotiky)]. V dalších podkapitolách budou jen v krátkosti zmíněny obě úlohy.

### 2.4.1 Přímá úloha

Spočívá v určení rovnice ramene robota, případně rychlosti a zrychlení zvolených bodů na základě známých souřadnic. Tyto souřadnice se nazývají *vnitřní souřadnice*, *kloubové souřadnice*, *strojové souřadnice* apod. Souřadnice se transformují do kartézského souřadného systému a v jednotlivých krocích se postupuje od základny robota po jeho koncový bod. Jednotlivé kroky se značí do tabulky podle následujících proměnných:  $\varphi$  - rotace podle osy  $z$ ,  $dz(e)$  - posun v ose  $z$ ,  $dx(a)$  - posun v ose  $x$ ,  $\alpha$  - rotace v ose  $x$ . Tyto proměnné je třeba dosadit do *transformační matice* (1) a vypočíst výslednou matici. Jsou-li souřadnice zadány jako funkce času, lze určit časový průběh veškerých poloh, natočení koncového nástroje, jejich rychlosti a zrychlení.

$$\mathbf{X}_i^{i-1} = \begin{pmatrix} \cos \varphi_i & -\cos \alpha_i \sin \varphi_i & \sin \alpha_i \sin \varphi_i & a_i \cos \varphi_i \\ \sin \varphi_i & \cos \alpha_i \cos \varphi_i & -\sin \alpha_i \cos \varphi_i & a_i \sin \varphi_i \\ 0 & \sin \alpha_i & \cos \alpha_i & e_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

### 2.4.2 Inverzní úloha

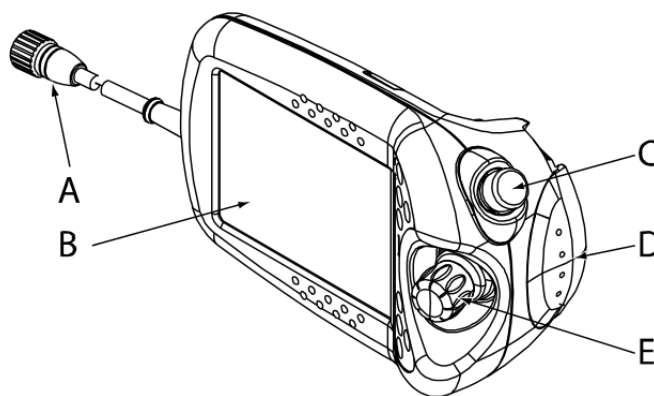
U inverzní úlohy je tomu poněkud jinak. Vycházíme z faktu, že známe pozici koncového členu manipulátoru. Potřebujeme nastavit jednotlivé klouby tak, aby robot dosáhl požadované pozice. Tato úloha má více řešení, v některých případech je nutno sáhnout k numerickému řešení, které zatěžuje řídicí systém robota. V případě robota se 6 stupni volnosti, je situace poměrně příznivá. Má-li robot sférické zápěstí, to znamená, že čtvrtá, pátá a šestá osa se protínají v jednom bodě, má robot vždy analytické řešení. V řídicím systému jsou dráhy



uloženy ve formě kloubových souřadnic. V samotném řízení se pohybují klouby v závislosti na čase tak, aby koncový bod robota udržoval danou pozici. Umožňuje vypočítat jednotlivá úhlová natočení všech os, kterými daná pozice a orientace bude dosažena. Přímá úloha nám pak slouží jako zpětná vazba. Ta totiž umožňuje určit z aktuálních natočení jednotlivých os robota polohu a orientaci koncového bodu.

## 2.5 Programování

Řídicí systém robota umožňuje několik druhů plánování, kde je vykonáván program. Programem je míněn sled akcí, které robot cyklicky provádí v navrženém algoritmu. Používané metody programování jsou popsány v následujících podkapitolách.



Obrázek 5: Dálkový ovladač

### 2.5.1 Dálkové (FlexPendant)

Pro tento účel je k řídicímu systému připojen konektorem (A) dálkový ovladač tzv. *FlexPendant*, který umožňuje nastavit jednotlivé body a trajektorie. Vše se programuje v bezpečném módu. V tomto módu je omezena rychlost kloubů a je zvýšená kolizní citlivost. Programování se provádí u starších systému pomocí funkčních kláves a u nového systému IRC5 pomocí aplikace na dotykové obrazovce (B). Robot se navádí do žádané pozice pomocí tříosého joysticku (E). Při tom lze robota navádět do pozice dvěma způsoby. Buď přímo nastavovat strojové souřadnice (úhlová natočení jednotlivých šesti os) nebo přímo řídit polohu a orientaci koncového členu vůči základnímu (nebo i jinému) tříosému kartézskému systému



robotu. Pro bezpečnost obsluhy, která robota programuje se využívá tzv. *Aktivační tlačítko* (D), to zamezí pohybu robota, pokud není správně stisknuté. Každý ovladač obsahuje tzv. *Central STOP* (C), ten při stisku okamžitě zastaví jakoukoliv činnost robota. Na obrázku č.5 je ukázka nejnovějšího FlexPendantu, určeného pro řízení robotů ABB. <sup>2</sup>

### 2.5.2 Offline (RobotStudio)

Výhodou offline programování je, že nepotřebujeme trávit veškerý čas se samotným řídicím systémem. Vše probíhá formou programování trajektorií a modelování mechanismů v softwaru, který spolupracuje s řídicím systémem. Při offline programování namodelujeme celou scénu, naprogramujeme robotická pracoviště a vše celé virtuálně odsimulujeme. Celý program se nahraje do řídicího systému a otestuje v reálné situaci, případně se program dopraví pomocí dálkového ovladače.



Obrázek 6: Ukázka komunikace

### 2.5.3 Online

Online programování je způsob, kterým můžeme při zaběhlé aplikaci sledovat či měnit program robota reálném čase. Tuto možnost podporují nejnovější řídicí systémy. Programování nejčastěji probíhá tak, že počítač obsahující řídicí software, je přímo spojen s řídicím systémem robota. Například software RobotStudio obsahuje virtuální FlexPendantu a mód online, který toto spojení umožňuje. Online programování je velmi vhodné do výrobních procesů, kde je potřeba zajistit nepřetržitou výrobu, bez nutnosti odstávky. S tímto způsobem se setkáme v procesech, které jsou řízeny kamerovým systémem nebo v aplikacích s proměnným prostředím. Zajišťuje se tak například manipulace předmětů na posuvný pás v potravinářském průmyslu.

<sup>2</sup>Obr. str.56, [www2.tec.ilstu.edu/students/ABB%20Robots/IRB140](http://www2.tec.ilstu.edu/students/ABB%20Robots/IRB140), (cz.pdf), read:(10.4.2013)



## 3 Pracovní stanoviště

Praktická část práce se odehrává v laboratoři Robotiky v přízemí budovy A, TU v Liberci. O rozvoj této laboratoře se zasloužil zejména Doc. Václav Záda, který se rozhodl vybudovat laboratoř robotiky jako nedílnou součást technické podpory výuky a výzkumu. Na základě projektu FRVŠ byly v roce 2004 zakoupeny tři průmyslové roboty od firmy ABB včetně řídicích systémů. Jedná se o roboty IRB140 a IRB1400 s řídicími systémy S4C+ a 1 ks repasovaného robota IRB1400 M97A s řídicím systémem S4C. Řídicí systém S4C je starší verze, která nepodporuje rozhraní Ethernet. Program lze nahrát do řídicího systému pomocí disket, jelikož S4C obsahuje disketovou mechaniku. Součástí dodávky bylo 12 ks vývojového software RobotStudio. Tento software umožňuje modelovat pracovní situace a simulovat je. V roce 2010 bylo zakoupeno čtvrté robotické rameno IRB120 s nejnovějším řídicím systémem IRC5. Dále bylo pořízeno několik menších mobilních robotů za účelem studování kolektivního chování robotů, tzv. *behaviour based control*.<sup>3</sup>

### 3.1 Automatizační technika ABB

Roboty jsou charakterizovány podle několika základních parametrů, které jsou uvedeny v produktové dokumentaci. Jedná se především o:

- **Pracovní oblast:** je parametr, který graficky udává oblast ve které se může pohybovat koncový člen robotu. Graficky znázorněná pracovní oblast je součástí obrázku č.4 na straně 18 . Jedná se o první část.
- **Dosah:** udává jak daleko dosáhne rameno od osy základny robota. Dosah je zobrazen také na obrázku č. 4 jako prostřední část.
- **Nosnost:** tento parametr udává maximální hmotnost tělesa, kterou můžeme připevnit na koncový člen robota. Graf zatížitelnosti je zobrazen na pravé části obrázku č.4. Pod nosnot také spadá zatížitelnost horního ramene robota, které se často využívá jako zasobník.

---

<sup>3</sup>Doc. Ing. Mgr. Václav Záda, CSc., Roboty na Fakultě mechatroniky, informatiky a mezipodoborových studií,  
URL: [www.fm.tul.cz/vaclav.zada/Laborator\\_Robotiky/Roboty\\_na\\_MTI-Zada.pdf](http://www.fm.tul.cz/vaclav.zada/Laborator_Robotiky/Roboty_na_MTI-Zada.pdf),read:(4.2.2013)



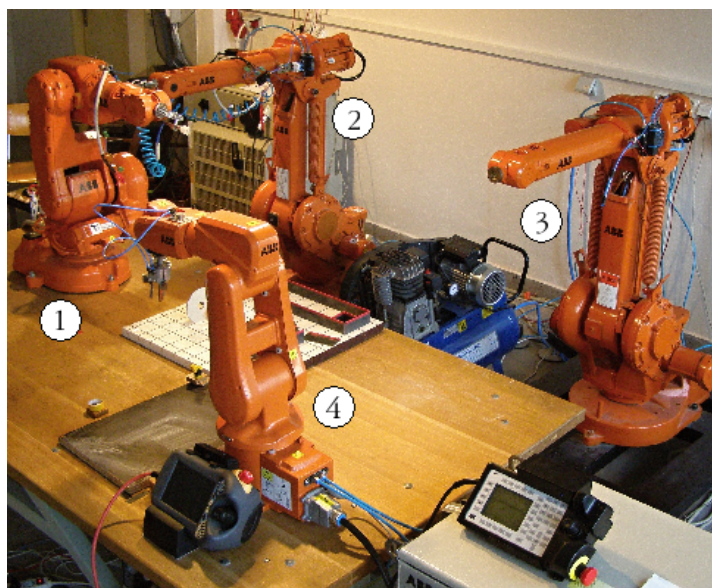
- **Opakovatelnost:** je přesnost při cyklickém opakování stejného pohybu. Běžně se udává v desetinách až setinách milimetrů. Je to údaj, který svědčí o kvalitě robotu daného výrobce.
- **Rychlost:** udává pohyb jednotlivý os robota ve stupních za sekundu. Tento údaj je však značně nepřesný, protože rychlost je udávána vlivem zrychlení a zpomalení. Především schopností tyto dva parametry řídit.
- **Přesnost:** je údaj, který udává jak se bude lišit trajektorie robota od té naprogramované. Nejčastěji vznikají problémy při prudkých změnách směru. Obecně platí, že se vzrůstající rychlostí robota se snižuje přesnost. Při tvorbě programu robota se proto zadává parametr *ZONEDATA*. Tento parametr slouží k udání velikosti odchylky od přesné trajektorie v *mm*. Více v podkapitole 4.1.3.
- **Stupeň krytí:** je parametr, který poukazuje na to, do kterého prostředí je roboty vhodné nasadit. Hodnota krytí je udávána parametrem IP. Jak se přiřazují hodnoty IP je možné najít například v knize Štěpána Berky [12, str. 229, kap. (Stupně ochrany krytí + nejvíce používané IP kódy)].

### 3.1.1 Roboty řady IRB

Každý robot v laboratoři robotiky na FM TUL, má své místo. Dva jsou připevněny na podvalech a dva na pracovním stole. To, na jakých místech jsou roboty umístěny, je vidět na fotografii č.7. Jednotlivé roboty jsou očíslovány pro snadnější odkazy.

- **1 - IRB140:** Kompaktní malý robot, který svou nosností do 5 kg, zvládá běžné průmyslové úkony. Jeho hlavní náplní je montáž, obloukové svařování, čištění, obsluha strojů nebo manipulace s materiálem. Je dodáván i s krytím IP67, takže je vhodný do náročnějších výrobních linek. Svojí opakovatelností 0,03 mm zajišťuje kvalitu pro přesné aplikace. Více v DataSheetu [10], kde jsou podrobné informace.
- **2 a 3 - IRB1400:** Spolehlivý střední robot, který zajišťuje nízké hladiny hluku a dlouhé servisní intervaly. Jeho hlavní využití je obloukové svařování. Na svém horním ramenu unese až 18 kg svařovacího drátu a jeho koncový bod zvládne manipulovat s předměty o váze 5 kg. Předností je dlouhý dosah. Další technické specifikace v DataSheetu [11].





Obrázek 7: Rozložení robotů na pracovišti

- **4 - IRB120:** Je to nejnovější přírůstek rodiny ABB. Jeho předností je kompaktnost, malé rozměry a váha. Je víceúčelový, takže ho najdeme u aplikací, kde stačí nosnost do 3 kg. A díky váze pouhých 25 kg je snadno přenositelný. Dosah ramene je 0,58 m. Podrobnější informace a technické specifikace je možné vyčíst z DataSheetu [9].

### 3.1.2 Řídicí systém

Samotný robot je bez řídicího systému bezvýznamný. Pro jeho správnou činnost potřebujeme kontroler, který vykonává instrukce zadané uživatelem ve formě programového kódu a díky výrobcem implementovaným algoritmům kinematiky a dynamiky mechanismu robotu, umožňuje přímo řídit polohu koncového bodu robotu. V laboratoři robotiky jsou k dispozici dva druhy řídicího systému. Jedná se o:

- **S4C:** toto je starší varianta řídicího systému. I tak je to plnohodnotný systém, který má vysokou úroveň spolehlivosti a může být použit v náročném prostředí. V laboratoři je k dispozici jeden systém S4C, který z důvodu problematického programování ve své práci nepoužívám. Využívám dvě novější verze systému S4C+ pro řízení robotu IRB140 a IRB1400. Technické specifikace řídicího systému udávané výrobcem jsou popsány v DataSheetu [15].





- **IRC5:** jedná se o nejnovější řídicí systém, který využívá společnost ABB. Konkrétně je v laboratoři umístěn *IRC5 Compact*, což je doposud nejmenší varianta tohoto systému. Je napojen na robota IRB120 a umožňuje přímou komunikaci s vývojovým softwarem RobotStudio. Podrobnější specifikace jsou uvedeny v dokumentaci [16].

### 3.2 Stavebnice ENGEL

Pro tvorbu efektorů a periferních zařízení je v laboratoři k dispozici montážní sada od Rakouské firmy ENGEL. Sada obsahuje nepřeberné množství komponent. Systém je především určen pro stavbu robotických chapadel. Jednotlivé prvky jsou správně dimenzované a jsou sladěné, aby se s nimi dobře manipulovalo. Základní kufr těchto komponent je zobrazen na obrázku č.8. Z této „stavebnice“ jsem setavil pro své potřeby dva efektory (*Prisavka1* a *Prisavka2*), které jsem následně usadil na roboty IRB140 (označeného jako 1) a IRB1400 (označeného jako 2 na obr.č.7). Podrobnější informace o sadě a další produkty lze nalézt na domovských stránkách firmy ENGEL. <sup>4</sup>



Obrázek 8: Stavebnice ENGEL

<sup>4</sup>[http://www.engelglobal.com/engel\\_web/global/de/911.htm](http://www.engelglobal.com/engel_web/global/de/911.htm) (read:12.3.2013)

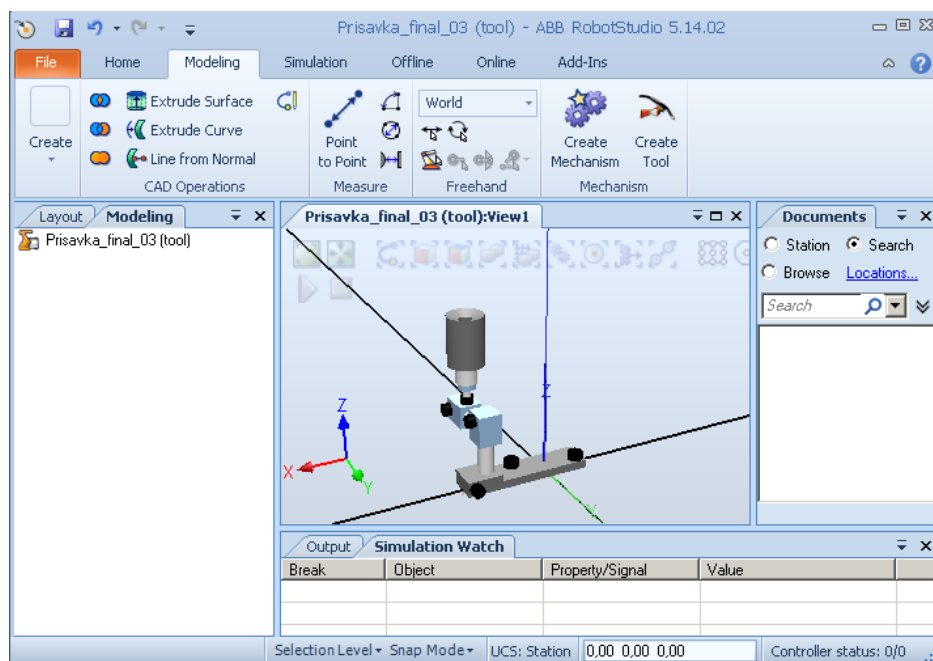


## 4 Software

Programové vybavení, které budu využívat při tvorbě aplikací bylo voleno s ohledem na fakt, že v naší laboratoři využíváme techniku od společnosti ABB. K technice je dodáván řídicí systém a software RobotStudio.

### 4.1 ABB RobotStudio

RobotStudio 5.14 je určen především k offline programování průmyslových robotů. Obsahuje přesný virtuální programovací model řídicího systému robotů páté řady. Software umožňuje vytvořit virtuálního robota a jeho řídicí systém přímo v počítači. V průmyslu je kladen důraz na to, aby byla většina nového kódu simulována virtuálně. Výhoda je v tom, že se nemusí tak často přerušovat běžící výroba. Software obsahuje mechanické části robotu i plný virtuální model řídicího systému se všemi funkcemi reálné jednotky. Výrobce poskytuje knihovny, které obsahují standardní zařízení běžně používaná v průmyslu. Jedná se o různé manipulátory, dopravníkové pásy a v neposlední řadě aktuální katalog robotů. Do prostředí Robot studio lze rovněž importovat jednotlivé prvky výrobní linky i vlastního manipulo- vaného či obráběného polotovaru ve standardních formátech CAD (*Computer aided design*).



Obrázek 9: Prostředí RobotStudio



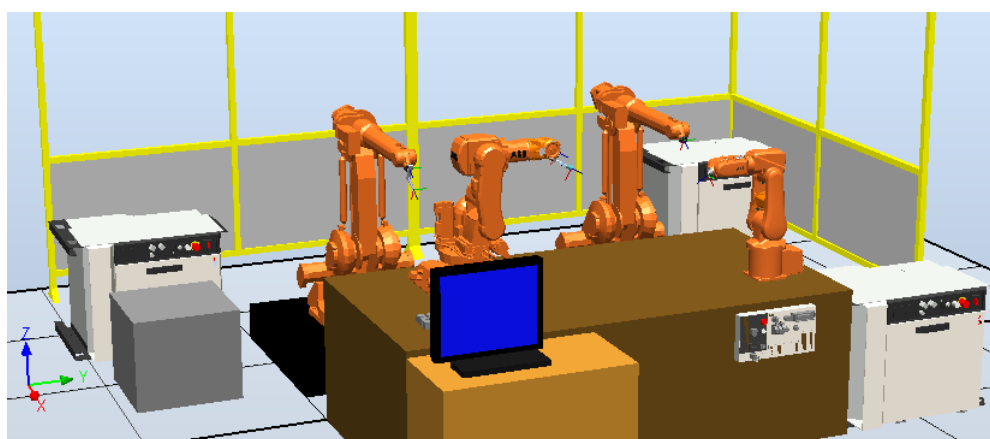
#### 4.1.1 Prostředí

Po prvním spuštění programu se zobrazí úvodní okno, na kterém jsou možnosti volby. Zvolíme *Open* pokud chceme otevřít uložený projekt nebo zvolíme *New / Create a new Station*. Je zde zobrazena možnost vytvoření prázdného stanoviště *Empty Station*, která se potvrdí ikonou *Create*. Nebo je možnost vytvořit stanoviště s virtuálním controlerem *Station with Robot Controler*.

Po volbě se dostaneme do hlavního okna RobotStudia, kde se nachází několik panelů. Hlavní okno je zobrazeno na obrázku č.9. Nacházejí se zde Karty *Home*, *Modeling*, *Simulation*, *Offline*, *Online* a *Add-Ins*, které obsahují panely nástrojů. Na levé straně se nachází základní navigace. V centru je modelovací prostředí *View*. V pravo je průzkumník, který spravuje vytvořené stanice. Ve spodní části se nachází okno s výpisem zpráv *Output*, *Watch*. Okno informuje o stavu překladači programu, výsledků simulace, režimu virtuálního systému apod.

#### 4.1.2 Virtuální laboratoř

Laboratoř jsem namodeloval po změření vzdáleností a rozměrů komponent, které se v ní nachází. Pro tvorbu předmětů jsem využil nástroj *Solid*, který obsahuje karta *Modeling*. Rozměry jednotlivých částí laboratoře jsou umístěny v následující tabulce č.1. Po vložení komponent, buď pomocí rozměrů z tabulky, nebo po načtení z knihovny *Home/Import Library*, získáme model virtuální laboratoře viz. obrázek 10.



Obrázek 10: Model laboratoře



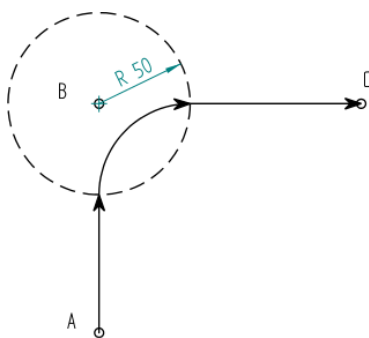
Tabulka 1: Souřadice modelů pracovního stanoviště

Název	Tvar	Rozměry	Poloha (x;y;z)	Rotace (x;y;z)
Stul	Kvádr	(1300; 2500; 810)	(0; -1000; 0)	(0; 0; 0)
Stul_PC	Kvádr	(500; 1000; 775)	(1300; -1000; 775)	(0; 0; 0)
Podval	Kvádr	(1150; 900; 2100)	(-1520; -1000; 0)	(0; 0; 0)
S4C+_1	ABB Library	(-; -; -)	(1300; 800; 810)	(0; 0; 90)
IRB140_C	ABB Library	(-; -; -)	(445; -330; 810)	(0; 0; 90)
IRB1400_1	ABB Library	(-; -; -)	(-950; -500; 210)	(0; 0; 0)
IRC5	ABB Library	(-; -; -)	(-1610; -1975; 350)	(90; 90; 90)
Stoleček	Kvádr	(600; 600; 502)	(-1100; -1930; 0)	(0; 0; 0)
PC_monitor	Kvádr	(60; 560; 460)	(1370; -900; 775)	(0; 0; 0)
PC_screen	Kvádr	(1; 500; 400)	(1430; -870; 805)	(0; 0; 0)
PC_klaves	Kvádr	(170; 450; 30)	(1500; -850; 775)	(0; 0; 0)
Stena_1	ABB Library	(-; -; -)	(-2760; 3000; 0)	(0; 0; 180)
Stena_2	ABB Library	(-; -; -)	(-2760; 3000; 0)	(0; 0; -90)
Stena_3	ABB Library	(-; -; -)	(-2760; 500; 0)	(0; 0; -180)
S4C+_2	ABB Library	(-; -; -)	(1350; 800; 510)	(0; 0; 90)
Podval_2	Kvádr	(1150; 900; 2100)	(-1520; -715; 0)	(0; 0; 0)
IRB1400_2	ABB Library	(-; -; -)	(-950; 1165; 210)	(0; 0; 0)
S4C	ABB Library	(-; -; -)	(-2000; 2100; 350)	(90; 90; 90)
Kostka 6ks.	Krychle	(37,5; 37,5; 37,5)	(volitelná)	(0; 0; 0)
Stojanek	Kvádr	(40; 30; 30)	(volitelná)	(0; 0; 0)
Prisavka1	Library	(-; -; -)	(Konc. bod IRB140)	(0; 0; 0)
Prisavka2	Library	(-; -; -)	(Konc. bod IRB1400)	(0; 0; 0)
Chapadlo	Library	(-; -; -)	(Konc. bod IRB120)	(0; 0; 0)

## 29



- **SPEED**: Je parametr, který udává jakou rychlostí se bude robot pohybovat na dané trajektorii. Nejmenší používaná rychlost se označuje **v5** a maximální rychlosti dosáhneme příkazem **vmax**. Jako standartní rychlost robota v automatickém režimu se používá příkaz **v1000**, což odpovídá rychlosti 1 m/s.
- **ZONEDATA**: Tímto parametrem se nastavuje s jakou přesností je třeba dosáhnout bodu jehož pozice je uložena v parametru Target. Myšleno jak přesně a pod jakým maximálním úhlem. Velká zóna se využívá u částí trajektorie, kde potřebujeme robotické rameno rychle přesunout bez ohledu na to, zda bodem projede přesně na milimetry. Naopak nulová zóna se zadává při konkrétní akci efektoru. Například při svařování je značně nevhodné mít tolerancí  $\pm 10$  mm. Na obrázku č.11 je zobrazena trajektorie o třech bodech, kde na bodě B je nastavena **ZONEDATA** jako **z50**.



Obrázek 11: Parametr ZONEDATA

- **TOOL**: Je základní informace o nástroji, se kterým robot pracuje při vykonávání instrukce. Tato hodnota je v základu nastavena jako **tool0**. *Tool* je proměnná, která se definuje v kalibračních datech programu. Důležitá je vlastnost, která v sobě nese polohu a orientaci koncového bodu efektoru tzv. *ToolData*. Je celkem běžné využívat systém rychlé výměny efektorů a proto je potřeba měnit i parametr Tool.
- **WORKOBJECT**: Považujeme za virtuální souřadný systém, který můžeme vztahovat k libovolnému bodu. Nejčastěji využíváme defaultní nastavení **WObj:=wobj0**, kde symbol „:=“ značí přiřazení. Pokud ovšem existuje skupina bodů, jejichž poloha se může vůči základně robota měnit. Je vhodné je přiřadit do nového *Workobjectu* a změnu polohy řešit pohybem celého souřadného systému.



## 5 Úlohy

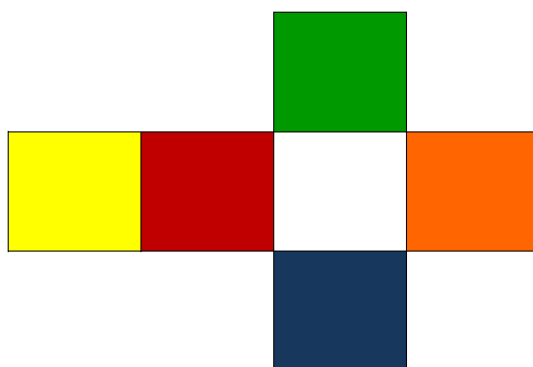
Jako ukázkové aplikace jsme s mým vedoucím zvolil tři úlohy, které demonstrují práci s průmyslovými roboty. Úlohy jsou předváděny na třech ze čtyř robotů. Cílem bylo demonstrovat opakovatelnost robota v automatickém režimu, spolupráce dvou průmyslových robotů a online programování za pomoci nejnovějšího řídicího systému.

### 5.1 Pyramida

Cílem této konkrétní aplikace je ukázat přesnost a opakovatelnost robota IRB140. Na koncový člen robota je upevněn efektor *Prisavka1* pomocí vyrobené redukce *Příruba*, který zajistí vycentrování efektoru, jednodušší montáž a neměnnost orientace přísavky při pohybu robota, která byla s předchozí verzí nedefinovatelná. Následně pak robot postupně odebírá dřevěné *Kostky* uložené v *Krabici*, která má přesně dané rozměry a rozteče pro jednotlivé kostky. Z odebraných kostek pak staví útvar podobný pyramidě. Výrobky, které jsem použil pro účel této aplikace, jsou popsány v následujících podkapitolách.

#### 5.1.1 Kostky

Hlavními předměty této úlohy jsou dřevěné kostky, které jsou vyrobeny na rozměr 375 mm. Celkem je použito devět kostek, z níž každá má polepené strany barevným papírem. Rozložení barev na kostce je zobrazeno na obrázku č.12. Pro vhodné uspořádání kostek je vyrobena úložná krabice s aretační podložkou z polystyrenu.



Obrázek 12: Barevné rozložení kostky





### 5.1.2 Úložná krabice a kalibrační podložka

Kostek je devět kusů a proto je potřeba zajistit pravidelné rozestupy, které mají kostky mezi sebou. Tento parametr zajišťuje úložná krabice, která je vyrobená z tvrdého milimetrového papíru a rozestupy, které mají kostky mezi sebou jsou vyplněny 1cm proužky polystyrenu. Krabice se skládá ze dvou kusů a to ze *Spodní části* a *Víčka*. Kompletní krabice vyplněná kostkami je zobrazena na obrázku č.13.



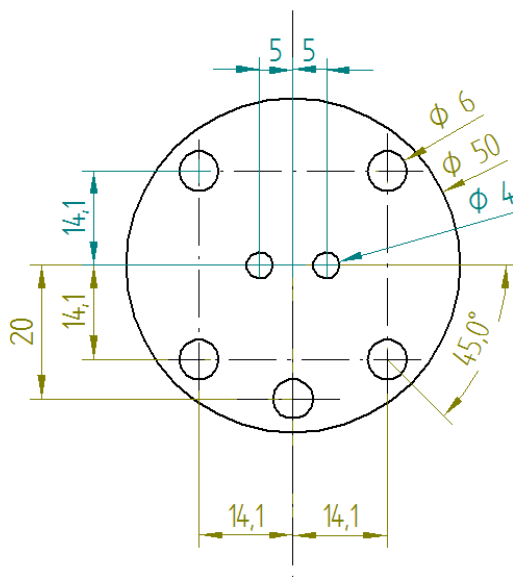
Obrázek 13: Krabice

### 5.1.3 Příruba

Způsob, kterým se v laboratoři přichytávají efekторы k robotu se mi zdál značně nevhodný, jelikož díly stavebnice ENGEL (kap.3.2) nejsou plně kompatibilní s koncovým členem robota. Komponenta stavebnice musela být převrtána, aby se docílilo alespoň nějakého přichycení ke koncovému bodu robota. Nehledě na to, že možností jak efektor posunout či orotovat bylo nepřehledné množství. Pro mé účely je nutné udržovat při přemísťování kosty z bodu A do bodu B předepsanou orientaci. Při manipulaci docházelo ke změně orientace kostky a k nesprávnému založení kostky, proto se mi tento způsob přichycení zdál značně nevhodný. V mých aplikacích potřebuji polohu a natočení efektoru aplikovat pokaždé stejně. Rozhodl jsem se navrhnout přírubu, která slouží jako mezičlánek mezi koncovým bodem robota a základnou efektoru.



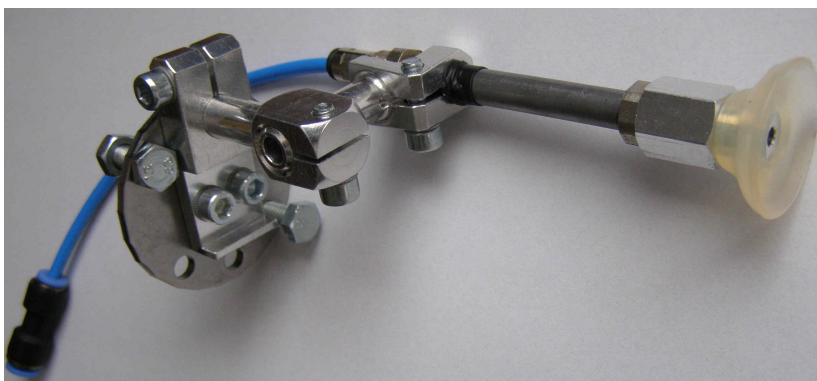
Výrobek je vysekán pomocí CNC vysekávacího lisu z 3mm plechu. Na povrchu jsou umístěny otvory, které jsou vyvrtány na totožných pozicích, jako je tomu u koncové člena robota. Navíc jsou však v přírubě dva otvory, které mají vytvořený závit M5. Pomocí těchto otvorů je sestavený efektor přišroubován k přírubě a poté pomocí čtyř otvorů přišroubován na koncový bod robota. Na obrázku č.14 je výkres výrobku vytvořený v parametrickém CAD softwaru pro 2D kreslení *SmartSketch*.



Obrázek 14: Spojovací příruba

#### 5.1.4 Prisavka1

Je efektor vytvořený částečně z dílu stavebnice ENGEL (kap.3.2) a částečně z kovové trubičky, která má na obou koncích vnitřní závit. Efektor je zvolen jako pevný a nepružný. Jediná deformace je povolena na na gumové části přísavky. Toto řešení je zvoleno, protože při použití pružné části ze stavebnice se přísavka otáčela kolem své osy a měnila se přitom zcela nedefinovaně orientace uchopovaného předmětu. Při prvních testech aplikace s pružnou přísavkou se totiž stávalo, že uchopená kostka se při přesunu mírně pootočila. Nejen že postavená pyramida vypadala nevzhledně, ale při následném vracení kostek zpátky do krabíčky se vyskytl problém a úloha se nemohla opakovat, jelikož by došlo k deformaci či ke zničení aretační podložky. Finální verze efektoru *Prisavka1* je i s přišroubovanou spojkou zobrazena na následujícím obrázku č.15.



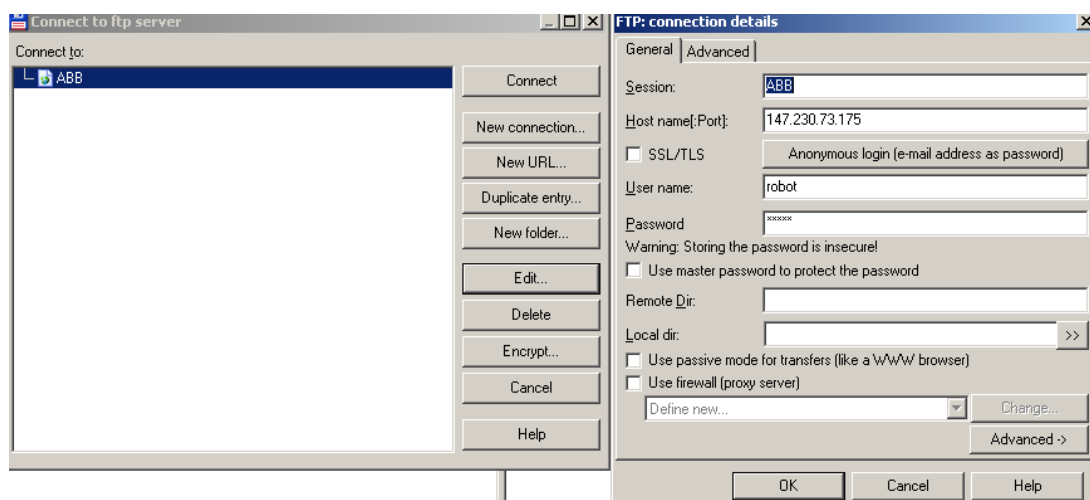
Obrázek 15: Přísavka pro IRB140

### 5.1.5 Program

Jak bylo zmíněno výše, řídicí program byl vytvořen v prostředí RobotStudio. Lze ho nalézt na příloženém CD. V této části nejprve popíšeme, jak vytvořený program nahrát do řídicího systému a následně jej spustit. Pokud v RobotStudiosu 5.14 (v záložce *Online RAPID language*) zvolíme *Save program as*, program se uloží standardně do třech souborů ve formátu *název\_programu.pgk*, *CalibData.mod* a *Module1.mod*. Tyto formáty nejsou plně kompatibilní se starší verzí řídicího systému S4C. Pro plnou kompatibilitu se starším systémem je potřeba spojit zdrojové soubory *CalibData.mod* a *Module1.mod* do jednoho kódu. Výsledný program je potřeba uložit tak, aby měl kompatibilní příponu *.prg*. Zvláštní pozor je potřeba dát na počet znaků v názvech Targetů, Workobjektů, atd., jelikož kontroler má problémy s dlouhými názvy a nepřeloží je.

Takto uložený program je možné nahrát do kontroleru robota, který jej bezproblému rozpozná. Pro komunikaci s řídicím systémem jsem zvolil rozhraní *Ethernet*. Po zapojení kabelu do síťového adaptéru je potřeba nastavit *FTP (File Transfer Protocol)* protokol pro připojení k řídicímu systému. Protokol nastavíme například v souborovém manažeru *Total Commander*. Ukázka nastavení protokolu je na obrázku č. 16. Zde se nastaví libovolný název připojení a komunikační port, který je **147.230.73.175**. Pro úspěšné připojení je potřeba zadat přihlašovací údaje (*User name*) a (*Password*) „robot“. Pokud je vše správně nastaveno stačí kliknout na tlačítko připojení (*Connect*) a vytvoří se nám spojení s daným kontrolerem. Program jednoduše přepokopírujeme do vybraného adresáře a spustíme na FlexPendantu. Nejprve je vhodné program otestovat vůči kolizím v ručním režimu a především zkalibrovat

s okolním prostředím. A poté je možné zvolit plně automatický režim, kde robot dosahuje rychlosti 1 m/s i více.



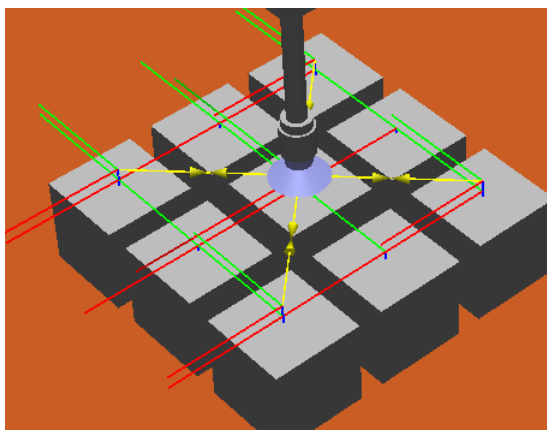
Obrázek 16: Nastavení FTP

V softwaru RobotStudio jsem nejprve vytvořil kalibrační program *KALIBRACE*, který je potřeba spustit v manuálním módu. Robot s výše popsaným efektem se postupně přesouvá směrem od středové kostky do všech směrů. Je potřeba krabíčku vhodně umístit a kostky které obsahuje patřičně zaaretovat. Tato kalibrace je nezbytná, jelikož pro účel demonstrace opakovatelnosti je potřeba zajistit co nejpřesnější polohu a orientaci úložné krabíčky. Respektive je potřeba zajistit, aby gumová část efektoru *Přísavka1* byla celou plochou na povrchu kostky. Při špatném nastavení by mohlo docházet k tomu, že by se přísavka nepřisála nebo že by při pokládání kostek zpátky do krabíčky došlo k deformaci kalibrační podložky. Měkký polyester je volen proto, že kdyby došlo ke kolizi, tak se nedeformuje efektor ale úložná krabice. Detailní zobrazení namodelované situace je zobrazeno níže na obrázku č.17.

Pokud je krabíčka na svém místě zaaretovaná, může se spustit hlavní sekvence programu *Pyramida*. Ten obsahuje dva podprogramy, které se cyklicky opakují. První z nich se nazývá *Postav*, ten má za úkol postupně odebírat kostky s krabíčky a následně z nich postavit pyramidu a druhý podprogram *Rozeber* pyramidu postupně rozloží a kostky přesune zpátky do krabíčky. Program se vykonává cyklicky do té doby, než nastane větší kolize, která by zabránila běhu motorů nebo do zásahu uživatele tlačítkem *Central STOP*. Je zde vidět opakovatelnost a přesnost offline programování. Pro pokusné účely byl vyzkoušen 10krát



opakovací cyklus, aby bylo dokázáno, že při správné kalibraci může program běžet samovolně s obrovskou přesností.



Obrázek 17: Kalibrování pozic

Během programování této úlohy, jsem využil zejména práci s tzv. *Workobjects*. *Workobject* je jakýsi souřadnicový systém, který můžu vytvořit například vůči souřadnému systému robota. Do tohoto systému potom umísťuji body (*Targety*). Řekněme například, že pyramidu chceme postavit někde jinde v dosahu robota. Posuneme tedy celý *Workobject* a sním i *Targety*, které ukazují na místa kde bude stát pyramida. Tento způsob je velmi efektivní, jelikož nezasahujeme do nadefinovaných bodů a neměníme tím nastavení os robota.

## 5.2 Spolupráce

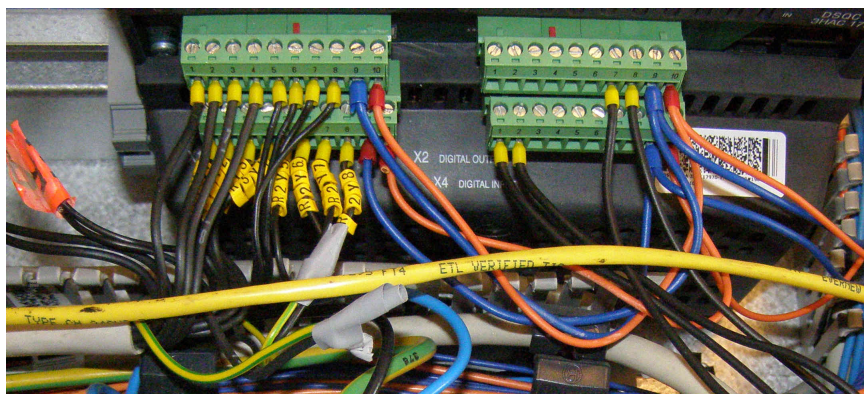
Tato úloha demonstruje možnost spolupráce průmyslových robotů. Především poukazuje na komunikaci, která je potřeba při předávání předmětů. V úloze pracuji s průmyslovými roboty IRB140 a IRB1400 (na fotografii č.7 se jedná o roboty č.1 a 2). Stejně jako v předešlé úloze budou roboty manipulovat s kostkami. Rozdíl je v tom, že roboty musejí při stavění pyramidy spolu spolupracovat. Při této spolupráci si musejí vzájemně vyměňovat informace. Výměna informací lze provést např. pomocí sériového rozhraní RS232 nebo digitálních vstupů a výstupů. Konkrétně využívám digitální vstupy a výstupy značené (1 - 8). Propojení je vidět na detailní fotografii č.18, kde je otevřená skříň řídicího systému robota IRB1400. Vzájemné propojení robotů lze využít při společné akci. Úloha je nastavená tak, aby jeden robot byl v roli „podavače“ a druhý v roli „odebírače“ kostek. Po sestavení libovolného



útvary z kostek se role obrátí a roboty vracejí kostky zpět na původní místo.

### 5.2.1 Výrobky

Při této úloze jsem použil některé výrobky z předchozí úlohy pro ulehčení postupu. Konkrétně se jedná o *Krabičku* a *Kostky*. Kostky s krabičkou budou plnit stejnou funkci jako v předchozí úloze. Krabička je základna ve které jsou kostky uloženy. Pozici úložné krabice lze opět nastavit kalibračním programem *KALIBRACE*. Také je zde využita *Příruba*, ovšem ve dvou provedeních, jelikož budeme využívat dva efektory. Použití příruby je stejné jako v minulé úloze. Je potřeba zajistit centralizaci a správné upevnění efektoru.

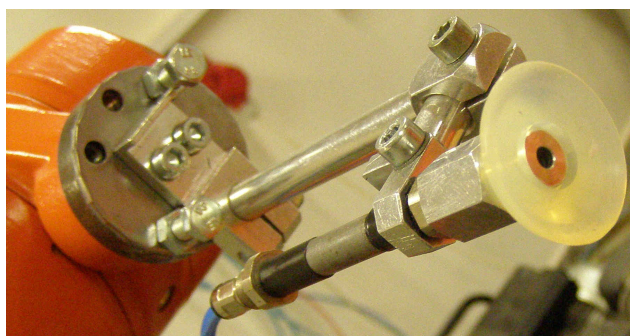


Obrázek 18: Detail propojení I/O

### 5.2.2 Efekторы

Na robota IRB140 umístím jako v předchozí úloze efektor *Prisavka1* a na robota IRB1400 připevním efektor *Prísavka2*, který je sestaven z podobných komponent stavbnice ENGEL. Je vhodné při montáži efektoru s přírubou dodržet orientaci, která je dána aretačním otvorem. Tento detail je vidět na fotografii č.19.

Na přísavkách je vytvořen podtlak, který ovládají elektromagnetické ventily firmy FESTO [17]. Tyto ventily jsou řízeny zapomocí digitálních výstupů. U IRB140 je to *DO10\_9* a u IRB1400 je to výstup *DO10\_16*. Do soustavy je neustále přiváděn tlak cca. 6 bar. Tlak zajišťuje kompresor, který je přes rozdělovač *Hoerbirger* přiváděn do jednotlivých robotů.



Obrázek 19: Detail Prisavka2

### 5.2.3 Program

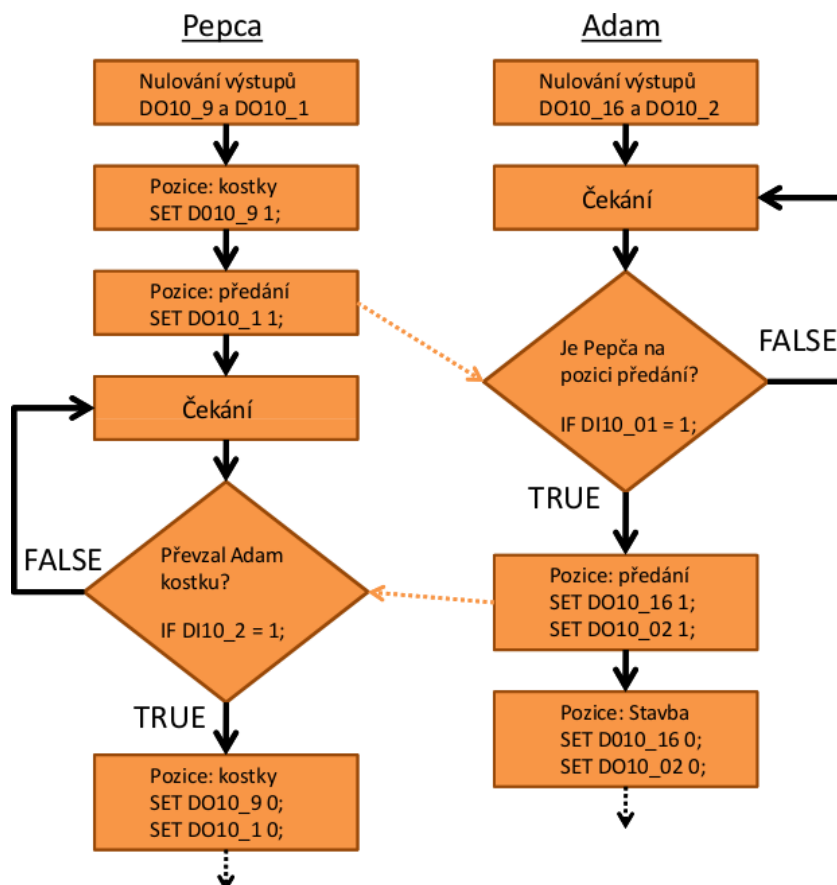
Nejprve je potřeba spustit na robotu IRB140 program *KALIBRACE*, který určí polohu krabičky s kostkami. Hlavní simulační program se nazývá *SPOLUPRACE* a obsahuje dvě části:

- **S\_***pepca*: tato část je určená pro řídicí systém robota IRB140. Obsahuje dva podprogramy s názvy *Podani* a *Odebrani*.
- **S\_***adam*: program je určen pro kontroler robota IRB1400. Taktéž obsahuje dva podprogramy s názvy *Odeber* a *Podej*.

Sekvence je naprogramovaná tak, aby robot IRB1400 (**adam**) čekal v nastavené pozici na příjezd IRB140 (**pepca**). Ten odebere kostku z krabičky a přesune se na místo předání. Když je pepca připraven, pošle pomocí digitálního výstupu *DO10\_1* signál. Ten vyhodnotí adam na digitálním vstupu *DI10\_01* a přesune svůj efektor na místo předání. Přisaje kostku a změní svůj digitální výstup *DO10\_02* tak, aby pepca zaznamenal na svém digitálním vstupu *DI10\_02* povel pro vypnutí elektromagnetického ventilu. To má za následek předání kostky. Pokud se tak stane, pepca se vrací ke krabičce pro následující kostku. Mezitím adam přesune kostku na vedlejší stoleček a staví z nich zvolený útvar. Jednoduchý diagram procesu předání kostky je zobrazen na obrázku č.20.

Přesnosti předávky se dá docílit vhodným nastavením Workobjectu *predavka*. Tento souřadný systém totiž obsahuje Targety místa předávky robota IRB1400. Pokud se tedy přísavky nestřetnou v místě předání, je potřeba souřadný systém posunout o případnou odchylku.





Obrázek 20: Diagram předání kostky

#### 5.2.4 Problémy

Při ladění programu v automatickém režimu se mi stávala zvláštní věc. V místě, kde si efekторы předávají kostku se vždy začínala pomalu vytvářet mezera. To způsobilo přibližně u čtvrté až šesté kostky pád. Nedokázal jsem zjistit příčinu toho, proč se od sebe efekторы vzdalují. Ošetřil jsem všechny pochybné možnosti v kódu, jako například nastavit menší rychlosti v místě předání a především nastavit parametr *ZONEDATA* na hodnotu **z0**, což by mělo zajistit přesné dosažení pozice oběma roboty v místě předávky. Problém přetrvával dále. S vedoucím práce jsme překalibrovali oba roboty do nulových poloh, jelikož tyto hodnoty byly mírně posunuté. Výsledek se nedostavil ani při pokusech s workobjekty.

Zbývala možnost důkladně vyměřit Target, který určuje místo předání a tyto data napevno vložit do základního Workobjectu **wobj0**. Výsledek byl takový, že se efekторы nepřestali vzdalovat. To znamenalo jediné, že se od sebe vzdalují samotní roboty. IRB1400 je připevněn na



podvalu, který má široké rozpětí a není možné s ním pohybovat. Zatímco IRB140 je přichycen k desce masivního stolu. Samotný stůl však k podlaze připevněn není. Robot v automatickém režimu vyvolává poměrně silné otřesy při náhlých změnách směru. Po bližším přezkoumání se ukázalo, že se stůl opravdu vzdaluje od podvalu a tím i od IRB1400. S tímto faktem se nedalo nic dělat, pokud by se ovšem neprovedlo rozsáhlé kotvení stolu. Řešením nakonec bylo rapidně snížit rychlost robota a do míst, kde docházelo k největším otřesům, umístit více *Targetů*. Tyto targety pomohou plynulosti pohybu ramene.

## 5.3 Tabule

Třetí úloha využívá nejmenšího a nejnovějšího robota IRB120 a jeho řídicí systém IRC5. Aplikace se chová tak, že robot pomocí tříbodového chapadla *FESTO HGDT* uchopí křídu, která je umístěná v kovovém stojánku. Pomocí této křídly kreslí na tabuli, umístěné na desce stolu. Úloha slouží k ukázce *Online* programování. Zkrze software RobotStudio programují kontroler robota, který řídí IRB120. Oproti předchozím úlohám je tento způsob programování velmi efektivní.

### 5.3.1 Stojan

Tato součást je vyrobena kovového kvádrů a zajišťuje stálou pozici křídě na pracovním stole. Křída je ve stojanu uložena ve svislé poloze, aby ji robot s tříbodovým efektem mohl bezpečně uchopit. Při vracení křídly zpět do stojanu upustí robot křídu mírně nad stojanem, aby nedošlo k deformaci křehké křídly. Křída bude do stojanu ukládána postupně i při kreslení, jelikož je potřeba regulovat ubytok materiálu. Robot s chapadlem „přehmátne“ křídu a může kreslit dál.

### 5.3.2 Chapadlo

Tato úloha se liší tím, že použitý efektor je daný. Jedná se o tříbodové chapadlo, umožňující uchopení předmětu podobně jako to zvládá lidská ruka. Hlavní využití tříbodových chapadel je manipulace s rotačními předměty. Technická specifikace je dostupná v datasheetu [18]. Toto chapadlo dokáže vyvinout sílu až 4 KN. To pro mé účely není zrovna vhodné. Křída se kterou budu manipulovat je křehká, a při tak velikém stisku by se mohla snadno

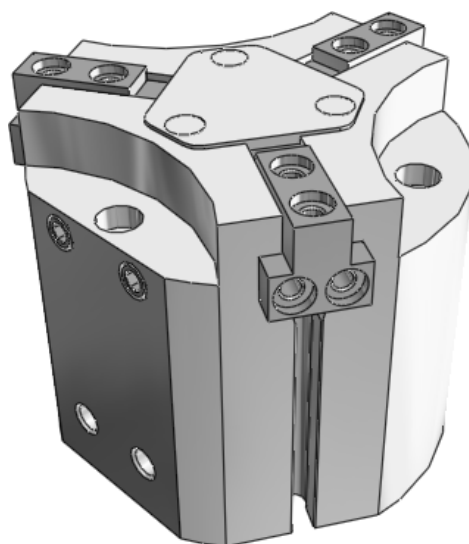




rozdrtit. Chapadlo jsem tedy vybavil pěnovými dorazy. Ty zabrání v drcení křídly. Tělo efektoru je zobrazeno na obrázku č.21. Model byl složen v programu AutoCad z dat, které poskytl výrobce.

### 5.3.3 Offline

Nejprve je potřeba určit přesnou pozici stojanu s křídou. To řeší kalibrační program *STOJAN*. Jednotlivé příkazy navádějí efektor a určují cílovou pozici stojanu. Další možností je upravit Workobject *wstojan* podle potřeby. Hlavní programová sekvence *Tabule* řeší pod jakým úhlem by měl být efektor nastaven a jak provádět jednotlivé tahy po tabuli.



Obrázek 21: Chapadlo FESTO

### 5.3.4 Online

Simulace online byla pouze vyzkoušena, ale z důvodu časové náročnosti při řešení výše popsaných problémů se již tuto úlohu nepodařilo zcela odladit na reálném systému.



## Závěr

Tato bakalářská práce byla vytvořena v sázecím systému  $\text{\LaTeX}$ . Snažil jsem se zde vytvořit aplikace v softwaru RobotStudio a následně je aplikovat na průmyslových robotech. První problém, na který jsem narazil, byla nefunkčnost licence RobotStudia mimo školní síť. Hodně času jsem tedy pracoval pod virtuálním operačním systémem Windows XP, jelikož zkušební verze softwaru je jen 30 dní. Snažil jsem se co nejčastěji navštěvovat laboratoř robotiky, abych mohl pracovat přímo u zdroje. Postupoval jsem tak, že jsem si namodeloval pracovní stanoviště, které je autentické s laboratoří. Vytvořil jsem modely různých efektorů, se kterými se v laboratoři běžně pracuje a testoval jsem jejich chování v reálné situaci. Měl jsem poněkud jinou představu o funkčnosti, a tak jsem si vytvořil efekторы vlastní. K nim jsem vyrobil i přírubu, pomocí které efektor upevňuji na koncový bod robota. Použil jsem dřevěné kostky se kterými v ulohách manipuluji.

Naprogramované aplikace jsou k dispozici na přiloženém CD. Jedná se o program *KALIBRACE*, který je potřeba pro správné nastavení pozice a orientace úložné krabice s kostkami. Program *Pyramida* je programován pro IRB140, ale může být aplikován i na jiných robotech. Program *SPOLUPRACE* je tvořen dvěma programy pro dva různé roboty. Ty spolu komunikují a spolupracují za účelem předávání kostek. Zde jsem řešil problém, se kterým jsem si delší dobu nevěděl rady. Efekторы se v místě předání od sebe vzdalovali. Vzdaloval se ovšem jeden z robotů. Díky prudkým pohybům v automatickém režimu posouval po inkrementech v řádu milimetrů se stolem. Poslední z programů je kalibrace *STOJAN*, která aretuje stojan s křídou pro aplikaci tabule. Program *TABULE* nebyl dokončen z důvodu zdržení se, při řešení výše popsaných problémů.

Při řešení bakalářské práce jsem měl možnost pracovat na velmi zajímavém projektu, který mi dal mnoho znalostí o robotice jako takové. Upřímně doufám, že výsledky práce budou užitečné a použitelné k výuce nebo k motivaci uchazečů usilujících o budoucí studium.



## Literatura

- [1] Rybička, J.: *L<sup>A</sup>T<sub>E</sub>X pro začátečníky*. Brno: Konvoj, 1999, ISBN 80-85615-42-8.
- [2] Karel Čapek: *Lidové noviny*. Novinový článek z data 24.12.1933.
- [3] Isaac Asimov: *Já, Robot Ivo Železný*, 1993, poč.str. 212, ISBN 80-7116-237-6.
- [4] Ing. Josef Černožanský, Ph.D.: *ABB Rapid language, RobotStudio*, přednáška předmětu Základy robotiky, TUL.
- [5] Záda, V.: *Robotika, matematické aspekty analýzy a řízení*, TUL, 2012, ISBN 978-80-7372-882-3.
- [6] Craig, J.J.: *Introduction to Robotics Mechanics and Control*, Third Edition, Prentice Hall, 2005.
- [7] *RAPID Reference Manual*, Article number: 3HAC 0966-13, Issue: For BaseWare OS 3.1 Rev.1, ABB Robotics Products AB, Sweden.
- [8] *Operating manual RobotStudio*, Document ID: 3HAC 032104-001, Revision: K, ABB Robotics Products AB, Sweden.
- [9] *IRB 120 Industrial Robot DataSheet*, file/ROB0149EN\_D\_LR.pdf, URL: <http://www.abb.com/product/seitp327/be2eef38406eaca4c125762000319182.aspx>
- [10] *IRB 140 Industrial Robot DataSheet*, file/PR10031EN%20R14%20LR.pdf, URL: <http://www.abb.com/product/seitp327/3939903968123b1fc125726f0043b847.aspx>
- [11] *IRB 1400 Industrial Robot DataSheet*, file/IRB%201410%20PR10325EN\_R3.pdf, URL: <http://www.abb.com/product/seitp327/56f6fd8b16df5b95c125726f0043cd35.aspx>
- [12] Štěpán Berka: *Elektrotechnická schémata a zapojení 2*, BEN - technická literatura, Praha 2010, ISBN 978-80-7300-254-1
- [13] Doc. Ing. Petr Blecha, Ph.D. a kol.: *Mechatronika Modul 10: Robotika*, Učebnice VÚT v Brně, EU – Projekt č. DE/08/LLP-LdV/TOI/147110



- [14] Doc. Ing. Zdeněk Plíva & Ph.D., Ing. Jindra Drábková, Ph.D.: *Metodika zpracování diplomových, bakalářských a vědeckých prací na FM TUL*, TUL, 2009, ISBN 978-80-7372-189-3
- [15] *S4Cplus Industrial Robot Controller* DataSheet, FILE/Datasheet+S4Cplus.pdf, URL: [www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/](http://www02.abb.com/global/nlabb/nlabb034.nsf/bf177942f19f4a98c1257148003b7a0a/068b10a5e5aa6ab1c1256f5b0047e00c/)
- [16] *IRC5 Compact controller*, DataSheet, file/ROBO150EN\_E%20LR.pdf, URL: <http://www.abb.com/product/seitp327/f0cec80774b0b3c9c1256fda00409c2c.aspx>
- [17] *Solenoid actuated valves*, DataSheet, Electronic valve, FESTO Pneumatic, URL: <http://www.festo.com/net/SupportPortal/Files/44028/JMEH-4-.pdf>
- [18] *3-point gripper HGDT-F*, DataSheet, file/info\_139\_en.pdf, FESTO Pneumatic, URL: [http://www.festo.com/cms/cs\\_cz/15573\\_15576.htm#id\\_15576](http://www.festo.com/cms/cs_cz/15573_15576.htm#id_15576)



## A Obsah CD

Soubor	Popis	Přípona
/library	Vytvořená knihovna obsahující modely	adresář
/station	Použitá pracovní stanoviště	adresář
/BP_hnyk	Elektronická podoba bakalářské práce	.pdf
/RAPID	Zdrojové kódy RobotStudio	adresář
/geometry	Geometrie vytvořených modelů	adresář
/systems	Použité virtuální kontrolery	adresář
/latex	Zdrojové kódy sázecího systému L <sup>A</sup> T <sub>E</sub> X	adresář
/video	Nahrávané simulace aplikací	adresář
/configuration	Nastavení virtuálních systémů	adresář